

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property  
Organization  
International Bureau



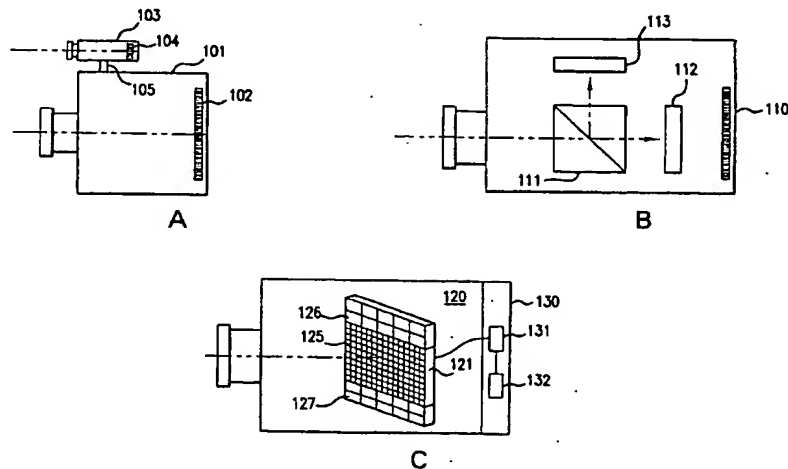
(43) International Publication Date  
31 December 2003 (31.12.2003)

PCT

(10) International Publication Number  
**WO 2004/001667 A2**

- (51) International Patent Classification<sup>7</sup>: **G06T**
- (21) International Application Number: **PCT/US2003/019446**
- (22) International Filing Date: **18 June 2003 (18.06.2003)**
- (25) Filing Language: **English**
- (26) Publication Language: **English**
- (30) Priority Data:  
**60/390,336** **21 June 2002 (21.06.2002)** **US**
- (71) Applicant (for all designated States except US): **THE TRUSTEES OF COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK [US/US]; 116th Street and Broadway, New York, NY 10027 (US).**
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **BEN-EZRA, Moshe [IL/IL]; Haarazim 12/26, Jerusalem (IL). NAYAR, Shree, K. [US/US]; 560 Riverside Drive, Apt. 15D, New York, NY 10027 (US).**
- (84) Designated States (regional): **ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).**
- (74) Agents: **TANG, Henry et al.; Baker & Botts, LLP, 30 Rockefeller Plaza, New York, NY 10112-4498 (US).**
- Published:  
— without international search report and to be republished upon receipt of that report
- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: **SYSTEMS AND METHODS FOR DE-BLURRING MOTION BLURRED IMAGES**



(57) Abstract: Systems and methods for providing a substantially de-blurred image of a scene from a motion blurred image of the scene are disclosed. An exemplary system includes a primary detector for sensing the motion blurred image and generating primary image information representing the blurred image, a secondary detector for sensing two or more secondary images of the scene and for generating secondary image information representing the two or more secondary images, and a processor for determining motion information from the secondary image information, estimating a point spread function for the motion blurred image from the motion information, and applying the estimated point spread function to the primary image information to generate information representing the substantially de-blurred image.

WO 2004/001667 A2

## SYSTEMS AND METHODS FOR DE-BLURRING MOTION BLURRED IMAGES

### CROSS-REFERENCE TO RELATED APPLICATIONS

This application is based on United States provisional patent  
5 application serial no. 60/390,336, filed June 21, 2002, which is incorporated herein by  
reference for all purposes and from which priority is claimed.

### NOTICE OF GOVERNMENT RIGHTS

The United States government has certain rights in the present  
10 invention pursuant to National Science Foundation ITR Award IIS-00-85864.

### BACKGROUND OF THE INVENTION

Technical Field. The present invention relates to techniques for  
15 digitally capturing and processing still images of a scene using an image detector, and  
more particularly, to techniques for correcting blurring introduced into such images  
due to motion of the image detector.

Background Art. Motion blur due to camera shake is a common  
problem in photography, especially in conditions involving zoom and low light.  
20 Merely pressing a shutter release button on the camera can in and of itself cause the  
camera to shake, and unfortunately cause blurred images. This problem is especially  
prevalent in digital photography, where lightweight cameras with automated exposure  
times are not easily stabilized are common, and where automated exposure times  
often necessitate relatively long stabilization to ensure a non blurred image. The  
25 compact form and small lenses of many of these cameras only serves to increase this  
problem.

The sensor of a digital camera creates an image by integrating energy  
over a period of time. If during this time – the exposure time – the image moves,  
either due to camera or object motion, the resulting image will exhibit motion blur.  
30 The problem of motion blur is increased when a long focal length (zoom) is  
employed, since a small angular change of the camera creates a large displacement of

the image, and in situations when long exposure is needed, either due to lighting conditions, or due to the use of small aperture.

There have been several attempts to provide a solution to this problem. One group of solutions focus on the reduction or elimination of relative movement between the camera and the scene during the integration time. Such solutions include the use of a tripod, flash photography, the use of increased sensor sensitivity, the use of an increased aperture, and dynamic image stabilization.

A stable tripod that can resist wind, and a shutter release cable that does not transmit hand vibration to a camera mounted on such a tripod, eliminates the problem of camera shake where both the mounted camera and scene are static. One limitation is that only professionals are likely to use a tripod and shutter release cable on a regular basis. Moreover, the use of a tripod does not solve the problem of shooting from a moving platform, such as car, train, helicopter or balloon.

A photographic flash produces a strong light flux that sustained for a fraction of a section (less than 1/1000). The exposure time is set to bracket the flash time (usually 1/60 sec), and the aperture of the camera is set to match the flash flux. Therefore, blur caused by motion during the bracket time has very low intensity. In essence, the flash "freezes" motion of both camera and moving objects. However, objects in bright daylight may still have motion blur and, of course, flash photography is useful only if the distance between the flash and the object is small.

Increasing the sensor sensitivity, and therefore reducing the exposure time, can decrease the problem of motion blur. However, it cannot eliminate blur completely. Moreover, Sensitive sensors (both film and CCD) produce noisy and grainy images.

Increasing the aperture size greatly decreases the required exposure time, and thus reduces motion blur. Unfortunately, cost and weight also significantly increase with an increased lens aperture, and a tripod may be required to comfortably handle such weight. Also, the use of a larger aperture lens is applicable only for more expensive cameras where it is possible to replace the lens.

In addition, the use of dynamic image stabilization involves the incorporation of inertial sensors, such as gyroscopes, to sense and compensate for camera shake in real time by moving an optical element. While this technology is used in stabilized zoom lens for Single Lens Reflex ("SLR") cameras, it is costly, and its effectiveness is limited to approximately 1/60 of a second for typical 400mm zoom

lens. The sensitivity of such system to very slow motion may also be limited, and may suffer from drift. In addition, such system cannot compensate for constant speed motion, such as occurs when taking images from a moving train.

Accordingly, while addressing the problem of motion of the camera  
5 itself is useful in certain applications, it does not provide an adequate solution to the problem of motion blur as such systems are either limited, very costly, or both. An alternative approach is to correct blur after the image has been taken by using a de-blurring algorithm.

However, while approaches which either assume that the point spread  
10 function is known or can be modeled by a simple function and found automatically from the image itself, no satisfactory solutions have thus far been provided. In particular, it has been difficult to obtain a useful point spread function useful in a de-blurring algorithm since inaccurate point spread functions tends to create strong artifacts, making them unpleasant for the eye. Accordingly, there remains a need for a  
15 technique for correcting blurring introduced into an image due to camera motion by finding an accurate point spread function.

#### SUMMARY OF THE INVENTION

An object of the present invention is to provide a technique for correcting blurring introduced into an image due to camera motion.

20 A further object of the present invention is to provide a technique for correcting blurring introduced into an image due to camera motion by using associated motion information.

Another object of the present invention is to provide an apparatus able to capture all information required to correct blurring introduced into an image due to  
25 camera motion.

Still another object of the present invention is to provide an apparatus for capturing a motion blurred image, de-blurring the image, and providing a user with a de-blurred image.

In order to meet these and other objects of the present invention which  
30 will become apparent with reference to further disclosure set forth below, the present invention discloses a system for providing a substantially de-blurred image of a scene from a motion blurred image of the scene. The system includes a primary detector for sensing the motion blurred image at a first predetermined resolution and generating

primary image information representing the blurred image, a secondary detector for sensing two or more secondary images of the scene and for generating secondary image information representing the two or more secondary images, and a processor. The processor is advantageously adapted to determine motion information from the secondary image information, estimate a point spread function for the motion blurred image from the motion information, and apply the estimated point spread function to the primary image information to generate information representing the substantially de-blurred image.

5 In one arrangement, the system includes a first camera housing the primary detector, a second camera housing the secondary detector, and a rigid member connecting the cameras. Alternatively, a single camera may house both the primary and secondary detectors.

In another preferred arrangement, a beam splitter having one input area and first and second output areas is provided. The beam splitter is optically coupled to the scene at the input area, to the primary detector at the first output area, and to the secondary detector at the second output area. Advantageously, the beam splitter may be an asymmetric beam splitter adapted to output greater than 50% of an input image energy through the first output area, and preferably approximately 90% of an input image energy through the first output area.

15 In still another preferred arrangement, the primary detector is a first portion of a dual-resolution sensor and the secondary detector a second portion of the dual-resolution sensor. The ratio of the first predetermined resolution to said second predetermined resolution is preferably 9:1 in terms of the scene energy incident on the sensor. The two portions may advantageously be formed on a single chip, to ensure a low cost and compact system.

20 The present invention also provides methods for providing a substantially de-blurred image of a scene from a motion blurred image of said scene. In one method, the motion blurred image of the scene and two or more secondary images are sensed. Next, primary image information representing the blurred image and secondary image information representing the two or more secondary images are generated, and motion information from the secondary image information is determined. A point spread function for said motion blurred image from said motion information; and the estimated point spread function is applied to the primary image information to generate information representing the substantially de-blurred image

Advantageously, fifteen or more secondary images of the scene should be sensed at the second predetermined resolution. It is preferred that global motion information is determined from the secondary image information, and a continuous point spread function estimated from the global motion information.

5           The accompanying drawings, which are incorporated and constitute part of this disclosure, illustrate preferred embodiments of the invention and serve to explain the principles of the invention.

#### BRIEF DESCRIPTION OF THE DRAWINGS

10           Figs. 1(a)-(c) are block diagrams of exemplary systems in accordance with the present invention;

Fig. 2 is a graph showing the relationship between temporal resolution and spatial resolution;

Fig. 3 is a graph showing an illustrative point spread function;

15           Figs. 4(a)-(d) are graphs illustrating the computation of a continuous point spread function from discrete motion vectors in accordance with an embodiment of the present invention;

20           Figs. 5(a) and (b) are a flow diagrams of a method for de-blurring a motion blurred image including deriving a point spread function in accordance with the present invention;

Figs. 6(a)-(d) are illustrative diagrams showing exemplary tools that may be used to model a point spread function in accordance with an alternative embodiment of the present invention;

25           Fig. 7 is a flow diagram of a method for determining a point spread function in accordance with the embodiment of Figs. 6(a)-(d);

Figs. 8 (a)-(b) are illustrative diagrams showing exemplary method for measuring a point spread function in accordance with another alternative embodiment of the present invention;

30           Fig. 9 is a graph of an exemplary estimated point spread function; and  
Figs. 10(a)-(c) are exemplary images of a scene.

Throughout the Figs., the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the present invention will

now be described in detail with reference to the Figs., it is done so in connection with the illustrative embodiments.

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

5 Referring to Figs 1(a)-(c), exemplary hybrid camera systems in accordance with the present invention are shown. Each system includes a primary image detector for capturing an image of the scene, as well as a secondary detectors for capturing information useful for correcting blurring introduced into the image due to camera motion.

10 The embodiment shown in Fig. 1(a) uses two cameras 101, 103 connected by a rigid member 105. Camera 101 is preferably a high-resolution still camera, and includes the primary detector 102 to capture an image of a scene. Camera 103 is preferably a low-resolution video camera which includes a secondary detector 104.

15 The secondary detector 104 is used for obtaining motion information, and therefore must capture a minimum of two frames of digital video information in order to provide such motion information. Preferably, fifteen or more frames are captured during the integration time of camera 101. While the embodiment shown with reference to Fig. 1(a) depicts a camera mounted on a camera, other two camera  
20 arrangements, such as a camera within a camera, may be utilized to achieve the same results.

The detector 102 may be traditional film, a CCD sensor, or CMOS sensor. Secondary detector 104 may likewise be a CCD or CMOS sensor. It is advantageous for the secondary detector 104 to be a black and white sensor, since  
25 such a detector collects more light energy (broader spectrum) and therefore can have higher temporal resolution. In addition, since the secondary detector is used only as a motion sensor; it can be of low spatial resolution to increase the temporal resolution and have high gain.

An alternative embodiment, shown in Fig. 1(b), employs a single  
30 camera 110 and a beam splitter 111 to generate two image paths leading to primary 112 and secondary 113 detectors. This system requires less calibration than the system of Fig. 1(a) since the same camera lens is shared, and hence results in identical image projection models.



Those skilled in the art will recognize that a tradeoff exists between the spatial resolution of the primary detector 112 and the ability to accurately provide motion information by the secondary detector 113 as the beam splitter 111 divides the available energy between the two detectors 112, 113. Therefore, while a beam splitter  
 5 which divides the energy 50/50 between the two detectors may be utilized, it is preferred that the division be greater than 80/20, for example, an approximately 90/10 split, with more energy going to the primary 112 detector.

In a highly preferred arrangement, the beam splitter 111 is an asymmetric beam splitter that passes most of the visible light to the primary detector  
 10 112 and reflects non-visible wavelengths toward the secondary detector 111. For example a "hot mirror" beam splitter which is commercially available from Edmund Industrial Optics may be employed.

Another alternative embodiment, shown in Fig. 1(c), uses a special chip 121 that includes both primary and secondary detectors. The chip 121 includes  
 15 both a high-resolution central area 125, which functions as the primary detector, and a low resolution peripheral areas 126, 127, which function as the secondary detector.

The chip 121 may be implemented using binning technology now commonly found in CMOS and CCD sensors. Binning allows the charge of a group of adjacent pixels to be combined before digitization. This enables the chip to switch  
 20 between a normal full-resolution mode, when binning is not used, and a hybrid primary-secondary detector mode, when binning is activated. In the hybrid mode, the primary detector portion 125 captures a high resolution image, while the secondary detector portions 126, 127 capture a sequence of low resolution images from which motion information can be derived.

Given sufficient light, CCD and CMOS sensors can detect a scene at  
 25 very fast rates and thereby avoid camera blur. However, motion blur will occur when there is not sufficient light for fast imaging, since the amount of energy reaching each pixel is  $\int \frac{\psi_{fov} \cdot k}{n} dt$ , where:  $\psi_{fov}$  is the flux through the field of view, k is the fill factor, n is the number of pixels, and the integral is taken over exposure time.  
 30 Reducing the number of pixels while keeping the same field of view equates into lowering resolution, and thereby increasing the energy per pixel.

Lower resolutions may be achieved either by using a low-resolution chip, or through binning, as discussed above. Examples for the image dimension of

the hi-resolution and low-resolution sensors for example ratios of  $\frac{1}{36}$  and  $\frac{1}{64}$  pixels at common resolutions are given in Table 1.

TABLE 1

Hi res	1024 x 768	1280 x 960	1600 x 1200	2048 x 1536	2560 x 1920
1/36 ratio	170 x 128	213 x 160	266 x 200	341 x 256	426 x 320
1/64 ratio	128 x 96	160 x 120	200 x 150	256 x 192	320 x 240

5 Also as shown in Fig. 1(c), the camera 120 preferably includes a circuit card or area which includes memory 121 for storing both the primary and secondary images sensed by detector portions 125-127. The camera also preferably includes processor 122 for computing motion from the sequence of low resolution images, estimating a point spread function for the primary image from such motion  
 10 information, and de-blurring the primary image with the estimated point spread function by applying a deconvolution algorithm, each of which are described below. The processor and memory should be sufficiently small to be implemented within the camera. Exemplary software that may be stored in memory 121 and executed on processor 122 is included herein as Appendix A. The de-blurred image may then be  
 15 displayed to the operator of camera 120 through a standard display (not shown), or stored for later use.

Alternatively, the motion computation, point spread function estimation, and de-blurring functions may be performed by a separate computer, such as a personal computer running the software of Appendix A. In addition, while the  
 20 foregoing description has been with respect to the embodiment shown in Fig. 1(c), it equally applies to the embodiments shown in Figs. 1(a) and (b), as each may be readily modified to include suitable memory and processing capacity. Likewise, the software of Appendix A is exemplary, and alternative software arrangements in a variety of programming languages may be utilized for performing such functionality.

25 Referring next to Fig. 2, a graph illustrating the fundamental tradeoff between spatial resolution and temporal resolution in an imaging system is shown. An image is formed when light energy is integrated by an image detector over a time interval. Assuming that the total light energy received by a pixel during integration must be above a minimum level for the light to be detected, the minimum level is  
 30 determined by the signal-to-noise characteristics of the detector. Therefore, given

such a minimum level and an incident flux level, the exposure time required to ensure detection of the incident light is inversely proportional to the area of the pixel. In other words, exposure time is proportional to spatial resolution. When the detector is linear in its response, the above relationship between exposure and resolution is also  
 5 linear.

The parameters of the line shown in Fig. 2 are determined by the characteristics of the materials used by the detector and the incident flux. Different points on the line represent cameras with different spatio-temporal characteristics. For instance, a conventional video camera 210 has a typical temporal resolution 30  
 10 fps and a spatial resolution of  $720 \times 480$  pixels. Instead of relying on a single point, two very different operating points on the line may be used to simultaneously obtain very high spatial resolution with low temporal resolution 220 and very high temporal resolution with low spatial resolution 230. This type of hybrid imaging provides the missing information needed to de-blur images with minimal additional resources.

15 Referring next to Fig. 3., an exemplary point spread function is shown. The complete point spread function of a motion-blurred image consists of two parts. First and most importantly, there is a point spread function due to motion. The derivation of such a point spread function is addressed in detail below. However, it should be noted that there may be a second component to a complete point spread  
 20 function, that of the imaging system itself, and may either be measured or modeled using a-priori knowledge of the imaging system. Those skilled in the art will appreciate that various techniques exist to conduct such measurement or modeling.

In order to determine the point spread function due to motion, a secondary detector provides a sequence of images (frames) that are taken at fixed  
 25 intervals during the exposure time. By computing the global motion between these frames, samples of the continuous motion path during the integration time may be obtained. The motion between successive frames is limited to a global rigid transformation model. However, the path, which is the concatenation of the motions between successive frames, is not restricted and can be very complex. Accordingly,  
 30 the motion between successive frames may be determined using a multi-resolution iterative algorithm that minimizes the following optical flow based error function:

$$\arg \min_{u,v} \sum \left( u \frac{\partial I}{\partial x} + v \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} \right)^2 \quad (1)$$

where the partial derivatives are the spatial and temporal partial derivatives of the image, and  $(u, v)$  is the instantaneous motion at time  $t$ . This motion between the two frames is defined by the following global rigid motion model:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & t_x \\ -\sin \theta & \cos \theta & t_y \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2)$$

where  $(t_x, t_y)$  is the translation vector and  $\theta$  is the rotation angle about the optical axis.

Note that the secondary detector, which has a short but nonzero integration time, may also experience some motion blur. This motion blur can violate the constant brightness assumption, which is used in the motion computation. However, under certain symmetry conditions, the computed motion between two motion blurred frames is the center of gravity of the instantaneous displacements between these frames during their integration time.

The discrete motion samples that are obtained by the motion computation need to be converted into a continuous point spread function. For this purpose, the constraints that a motion blur point spread function must satisfy are defined and then used in order to estimate the appropriate point spread function.

Any point spread function is an energy distribution function, which can be represented by a convolution kernel  $k : (x, y) \Rightarrow w$ , where  $(x, y)$  is a location and  $w$  is the energy level at that location. The kernel  $k$  must satisfy the following energy conservation constraint:

$$\iint k(x, y) dx dy = 1 \quad (3)$$

which states that energy is neither lost nor gained by the blurring operation ( $k$  is a normalized kernel). In order to define additional constraints that apply to motion blur point spread functions, a time parameterization of the point spread function is used as a path function  $f : t \Rightarrow (x, y)$  and an energy function  $h : t \Rightarrow w$ . Due to physical speed and acceleration constraints,  $f(t)$  should be continuous and at least twice differentiable, where  $f'(t)$  is the speed and  $f''(t)$  is the acceleration at time  $t$ .

By assuming that the scene radiance does not change during image integration, an additional constraint is determined:

$$\int_t^{t+\delta t} h(t) dt = \frac{\delta t}{t_{\text{end}} - t_{\text{start}}}, \delta t > 0, t_{\text{start}} \leq t \leq t_{\text{end}} - \delta t, \quad (4)$$

where  $[t_{\text{START}}, t_{\text{END}}]$  is the image integration interval. This constraint states that the amount of energy which is integrated at any time interval is proportional to the length of the interval.

Given these constraints and the motion centroid assumption, a continuous motion blur point spread function may be estimated from discrete motion samples, as illustrated in Figs. 4 (a)-(d).

First, the path  $f(t)$  may be estimated by Spline interpolation, as shown in Figs. 4(a) and (b). Spline curves are preferably used because of their smoothness and twice differentiability properties, which satisfy the speed and acceleration constraints.

In order to estimate the energy function  $h(t)$ , the extent of each frame along the interpolated path must be determined. This may be accomplished using the motion centroid assumption by splitting the path  $f(t)$  into frames with a Voronoi Tessellation, as shown in Fig 4(b).

Since the constant radiance assumption implies that frames with equal exposure times integrate equal amount of energy,  $h(t)$  may be computed, up to scale, for each frame as shown in Fig. 4(c). Note that all the rectangles in this figure have equal areas.

Finally,  $h(t)$  is normalized in order to satisfy the energy conservation constraint and smooth it. The resulting point spread function is shown in Fig. 4(d). The end result of the above procedure is a continuous motion blur point spread function that can now be used for motion de-blurring.

Given the estimated point spread function, the high-resolution image that was captured by the primary detector may be de-blurred using well known image deconvolution algorithms, such as the Richardson Lucy algorithm. Since this is the only step that involves high-resolution images, it dominates the time complexity of the method, which is usually the complexity of a Fast Fourier Transform ("FFT").

Referring next to Fig. 5(a), the foregoing techniques are implemented in a methodology as follows. First, primary image information representing the blurred image sensed by the primary detector 510, and secondary image information representing a sequence of images sensed by the secondary detector 511, are obtained. A standard motion analysis algorithm is used to determine discrete motion information 520 of the primary detector. Next, the point spread function for the motion blurred image is estimated 530 using the discrete motion information. That

point spread function 535 may optionally be convolved with an estimated or measured point spread function for the optical system itself 540. The point spread function is then applied to the primary image information in a standard de-blurring algorithm, 550, and a de-blurred image is output 560.

5           Fig. 5(b) shows the preferred details of the point spread function estimation step 530. Two-dimensional Spline interpolation is used to provide a continuous two-dimensional path 531. Voronoi Tessellation is then used to provide frame partitioning of the interpolated two-dimensional path 532. Equal area rectangles are constructed within each partitioned frame to determine the mean energy  
10       at each frame 533. Finally, the determined mean energy values are smoothed and normalized 534.

Referring next to Figs. 6(a)-(d), exemplary tools that may be used to model a point spread function in accordance with an alternative embodiment of the present invention are shown. In some cases, a point spread function can be estimated  
15       directly from an image itself, without the need for additional motion information. For example, a small bright point light source on a dark background, such as dark night, which happen to be at the right depth, if the camera was translating, or if the point light source was at arbitrary depth (bright star in clear sky) and camera was rotating with no translation – then the image of this point light source provide point spread  
20       function which is good enough for de-blurring, as long as the dynamic range of the camera is sufficient.

It is unlikely to expect such luck to happen, especially if the camera motion included translation about the optical axis, since at least two such points are needed. Instead, a set of primitives may be created that, if found in the image, can  
25       help estimating the point spread function. The greatest advantage of this approach is that it does not require any additions to the imaging process itself and it can be used for existing pictures as well. The disadvantage is that this method relies on user skills to estimate the real shape of an object from a blurred image – or from a different image that may not be blurred that was taken at a different time or different angle.

30           Fig 6(a) illustrates an exemplary point tool 610 that may be used to define an object point. Since object points may not be perfect small white points over a black background, the tool provides means to define point size, eccentricity, orientation and color. Point size 611 is selected by a slider or by entering size in pixels of a fraction of a pixel. Point eccentricity 612 is selected by a slider or entered

as a number as the ratio between main axes. Regarding orientation 613, if point eccentricity is not 1, then the orientation can be entered using a dial or as a number (angle). Point color 614 is selected using color tool, or sampled from the image itself and optionally modified. In addition, a background color may be selected using color tool, or sampled from the image itself and optionally modified.

Fig 6(b) illustrates an exemplary line tool 620 that may be used to define an object line. The line attributes may include thickness, orientation, and color. Line thickness 621 is selected by a slider, or by entering thickness in pixels (can be fraction of a pixel). Line orientation 622 can is entered using a dial or as a number (angle). Line color 623 is selected using color tool – or sampled from the image itself and optionally modified. Again, a background color may be selected using color tool, or sampled from the image itself and optionally modified.

Fig 6(c) illustrates an exemplary ramp tool 630 that may be used to define a ramp or edge. The attributes may include orientation and color. Ramp orientation 631 can is entered using a dial or as a number (angle). Ramp color 63 may be selected using color tool, or sampled from the image itself and optionally modified. A background color may be selected using color tool, or sampled from the image itself and optionally modified.

Fig 6(d) illustrates an exemplary corner tool 640 that may be used to define an object corner. Corner attributes include angle, orientation, and color. The angle 641 may be entered using a dial or as a number (angle). Corner orientation 642 can is entered using a dial or as a number (angle). Corner color 643 may be selected using color tool, or sampled from the image itself and optionally modified. A background color may be selected using color tool, or sampled from the image itself and optionally modified.

For example, a user may download a motion-blurred image from the Internet, and desire to de-blur that image. Using the corner tool of Fig. 6(d), the user may examine a small region in the blurred image, e.g., 30x30 pixels, and create a model 30x30 pixel image of what that region should look like when de-blurred. That model image, convolved by an unknown point spread function, will equal the original region of the blurred image.

One approach to finding this block is by using a Fourier transform. The Fourier transform of the model image region multiplied by the Fourier transform of the point spread function is equal to the Fourier transform of the captured image

region. Therefore, the Fourier transform of the point spread function is determined by dividing the Fourier transform of the captured image region by the Fourier transform of the model image region, and an inverse Fourier transform may be used to obtain an estimated point spread function of the blurred image. Once obtained, the user may  
5 de-convolve the blurred image with the estimated point spread function to obtain an estimate of the captured image. The user then can then compare the estimated captured image with the original captured image, and visually determine whether the further refinements are necessary.

Referring next to Fig. 7, a method for interactively estimating a point  
10 spread function using the tools of Figs. 6(a)-(d) is shown. A user selects and classifies features within the blurred image 720, e.g., using the tools of Figs. 6(a)-(b). A common point spread function is then determined using de-convolution, where the function is the unknown variable 730. The motion blurred image is de-blurred using the recovered point spread function 740, and the user is permitted to view the de-  
15 blurred image and refine his or her classifications as appropriate 750, 755. Finally, the de-blurred image is output 760.

Referring next to Figs. 8(a)-(b), an exemplary method for measuring a point spread function in accordance with another alternative embodiment of the present invention will be described. Laser guides are common in adaptive optics to  
20 provide a reference point for wavefront measurements. Similar techniques may be useful for motion de-blurring by projecting a reference point on the object using a laser beam as seen in Fig. 8(a). An alternative approach is to attach a (limited) stabilized laser to the camera itself as shown in Fig. 8(b).

As shown in Figs. 8(a) and (b), a laser guide is attached to a camera.  
25 The laser is mounted on rings (gimbals), which are stabilized using gyroscopes. The laser emits one or more reference points to the scene. The image of these points is then utilized, after normalization to satisfy an energy constraint, to determine the point spread function that is sought. It should be noted that other light beam sources, such as columnated light beam sources, may be used in place of a laser.

30 It should be noted that the practical use of this embodiment is limited, as transmitting a laser beam into living objects may not be practical. However, the technique is useful for natural or commercial photography.

Fig. 9 is a graph of an exemplary estimated point spread function derived using the apparatus of Fig. 1(a) and the method described in connection with



Fig. 8. Figs. 10(a)-(c) are exemplary images of the corresponding scene, which Fig. 10(a) showing an image of the scene taken from a tripod-mounted camera, Fig. 10(b) showing the blurred image, and Fig. 10(c) showing the image after de-blurring. While the de-blurred image reveals some artifacts from the de-blurring process, it is a  
5 vast improvement over the blurred image. Accordingly, techniques for correcting blurring introduced into an image due to camera motion have been provided.

The foregoing merely illustrates the principles of the invention. Various modifications and alterations to the described embodiments will be apparent to those skilled in the art in view of the teachings herein. For example, the foregoing  
10 techniques may be applied to video sequences taken by a moving camera, using an assumption of symmetrical distribution to recover the average motion vectors from the motion blurred images. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within  
15 the spirit and scope of the invention.

Appendix A

```
// Columbia University. Cave Lab.
```

```
#include <image.h>
#include <pnm__env.h>
```

```
#include "pyramid.h"
#include "motion.h"
#include "reg.h"
```

```
int main (int argc, char *argv[])
```

```
fMonoImg img1, img2;
fMotion m;
```

```
if (argc < 4) {
    fprintf (stderr, "%s - register two (well cropped) images.\n");
    fprintf (stderr, "Usage: %s <img1> <img2> <model>\n", argv[0]);
    fprintf (stderr, "        <img1>      - First image to register.
    Type PPM or PGM\n");
    fprintf (stderr, "        <img2>      - Second image to
    register. Type PPM or PGM\n");
    fprintf (stderr, "        <model>     - Motion model 't' for
    translation 'r' for rotation
    return 1;
}
```

```
img1 = pnm(argv[1]);
img2 = pnm(argv[2]);
```

```
if (*argv[3] == 'r')
    m = img_register (img1, img2, ROTATION);
else
    m = img_register (img1, img2, TRANSLATION);
```

```
m.print(stdout);
```

```
return 0;
```

```
theta = var[0];  
  
res.a = cos(theta);  
res.b = - sin(theta);  
res.c = - res.b;  
res.d = res.a;  
  
res.tx = var[1];  
res.ty = var [2];  
  
return res;
```

```
// Columbia University. Cave Lab.

//=====
//
// Name: reg.h - image registration header file
//
//=====

#ifndef REG_H
#define REG_H

//=====
// function prototype
//=====

void unwarp (fMonoImg &dst, fMonoImg &src, fMotion &m);

fMotion img_register (fMonoImg &img1, fMonoImg &img2, int mType);
fMotion pyr_register (fMonoPyr &I1, fMonoPyr &I2, fMonoPyr &Ix, fMonoPyr &Iy,
int
fMotion motion_translation (fMonoImg &Ix, fMonoImg &Iy, fMonoImg &It);
fMotion motion_rotation (fMonoImg &Ix, fMonoImg &Iy, fMonoImg &It);

#endif
```

```
unsigned GetLevels()    {  
    return levels;  
}  
  
~Pyramid( ) {  
    for (unsigned i=0; i<levels;i++)  
        data [i] = T ( );           // null assignment frees data  
    delete [] data;  
};  
  
typedef Pyramid<fMonoImg>    fMonoPyr;  
typedef Pyramid<fRgbImg>     fRgbPyr;  
  
#endif
```

```

// Columb_a University Cave Lab.
// June 2002

//=====
// Motion.h - affine motion class
//=====

#ifndef MOTION_H
#define MOTION_H

#define Epsilon (1.0e-8)

#define TRANSLATION 1
#define ROTATION 2

class fMotion {

public:

    double a, b, tx;          // affine parameters
    double c, d, ty;

//=====
// ctor
//=====

    fMotion ( ) {
        a = d = 1.0;
        b = c = 0.0;
        tx = ty = 0.0;
    }

//=====
// Multiply motion
//=====

    fMotion mul (const fMotion &m) {
        fMotion res;

        res.a = a*m.a + b*m.c;
        res.b = a*m.b + b*m.d;
        res.tx = a*m.tx + b*m.ty + tx;

        res.c = c*m.a + d*m.c;
        res.d = c*m.b + d*m.d;
        res.ty = c*m.tx + d*m.ty + ty;

        return res;
    }
}

```

```
//=====
// Inverse motion
//=====
```

```
fMotion inv ( ) {
```

```
    fMotion res;
    double t4;
```

```
    t4      = (- a*d+c*b);
```

```
    if (fabs(t4) < Epsilon) {
```

```
        src.SetOrigin(src.GetWidth()/2,src.GetHeight()/2);
        dst.SetOrigin(dst.GetWidth()/2,dst.GetHeight()/2);
```

```
        unwarp(dst, src, m) ;
```

```
        pnm((const bRgblmg) dst).Write(argv[2]);
```

```
        return 0;
```

```
    }
```

```

// Columbia University. Cave Lab.

//=====
//
// reg. cpp - image registration implementation
//
//=====

#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#include <math.h>

#include "image.h"
#include "pnm_env.h"

#include "motion.h"
#include "image.h"
#include "pyramid.h"
#include "reg.h"
#include "solve.h"

#define MIN(x,y) ((x) < (y) ? (x) : (y))

//=====
//
// unwarp - unwarp image by motion
//
//=====

void unwarp (fMonoImg &dst, fMonoImg &src, fMotion &m)
{
    int    x,y;
    float nx, ny;

    int startX, stopX, startY, stopY;

    startY = dst.GetYmin();
    stopY = dst.GetYmax();

    startX = dst.GetXmin();
    stopX = dst.GetXmax();

    for (y=startY; y<stopY; y++){
        for (x=startX; x<stopX; x++){

            nx = (float) (m.a * x + m.b * y + m.tx);
            ny = (float) (m.c * x + m.d * y + m.ty);

            dst.Pix(x,y) = src.bPix(nx,ny);
        }
    }
}

```



```
}  
}
```

```
/=====   
/ img_register - register two monochrome images   
/=====
```

```
Motion img_register (fMonoImg &img1, fMonoImg &img2, int mType)
```

```
// error_msg handlers.
```

```
#include "stdio.h"  
#include "error_msg.h"
```

```
static char buf [256];
```

```
void error_msg (char *s, int p)  
{  
    fprintf(stderr,s,p);  
}
```

```
// int exception may have scope that is too large
```

```
void clip_exception(int x)  
{  
    throw x;  
}
```

```

fMotion res;

int levels, size;
int i;

size = MIN(img1.GetWidth( ), img1.GetHeight( ));

levels = (int) (log((double) size) / log(2.0)) - 4;

//limit levels
if (levels < 1) levels = 1;
if (levels > 6) levels = 6;

fprintf(stderr, "Levels: %d\n", levels);

fMonoPyr I1(img1, levels);
fMonoPyr I2(img2, levels);

// create empty pyramid
fMonoPyr Ix(levels);
fMonoPyr Iy(levels);

// derive with kernel of 5
for (i=0; i<levels; i++){

    Ix[i] = I1[i].dx(5);
    Iy[i] = I1[i].dy(5);
}

fMonoImg img3(img1.GetWidth( ), img1.GetHeight( ));

// set origin to center

img1.SetOrigin(img1.GetWidth( )/2, img1.GetHeight( )/2);
img2.SetOrigin(img2.GetWidth( )/2, img2.GetHeight( )/2);
img3.SetOrigin(img3.GetWidth( )/2, img3.GetHeight( )/2);

for (i=0; i<levels; i++) {
    I1[i].SetOrigin(I1[i].GetWidth( )/2, I1[i].GetHeight( )/2);
    I2[i].SetOrigin(I2[i].GetWidth( )/2, I2[i].GetHeight( )/2);
    Ix[i].SetOrigin(Ix[i].GetWidth( )/2, Ix[i].GetHeight( )/2);
    Iy[i].SetOrigin(Iy[i].GetWidth( )/2, Iy[i].GetHeight( )/2);
}

res = pyr_register(I1, I2, Ix, Iy, mType);

unwarp(img3, img2, res);

pnm ((const bMonoImg) img3). Write ("wpr.pgm");

return res;

```

```

/=====
/ pyr_register - register two monochrome pyramids
/=====

```

```

Motion pyr_register (fMonoPyr &I1, fMonoPyr &I2, fMonoPyr &Ix, fMonoPyr &Iy,
int

```

```

{
    fMonoImg It;
    fMotion res, cur, tmp;

    int lvl, levels, no_motion, max_iter;
    int i;

    levels = I1.GetLevels( );

    max_iter = 50;

    // copy I2 as ID warp

    fMonoPyr I2w(levels);

    for (i = 0; i < levels; i++) {
        I2w[i] = I2[i];
    }

    for (lvl = levels-1; lvl >= 0; lvl--) {
        for (i = 0; i < max_iter; i++) {

            It = I2w[lvl] - I1[lvl];

            switch (mType) {

            case TRANSLATION :
                cur = motion_translation (Ix[lvl], Iy[lvl], It);
                break;

            case ROTATION :
                cur = motion_rotation (Ix[lvl], Iy[lvl], It);
                break;

            default:
                assert(0);                // get here only due to a Bug !!
                break;
            }
            cur.tx /= 2;
            cur.ty /= 2;

            res = res.mul(cur);

            //in place unwarp for next iteration
            unwarp(I2w[lvl], I2[lvl], res);

```

```

    }
    fprintf(stderr, "\nLevel: %2d Res: %03d x %03d\n", lvl, Ix[lvl].GetWidth(), Ix[lvl].GetHeight());
    tmp = res;
    tmp.tx *= (1 << lvl);
    tmp.ty *= (1 << lvl);
    tmp.print(stderr);

    res.tx *= 2;
    res.ty *= 2;

    //warp for next level
    if (lvl > 0)
        unwarp(I2w[lvl-1], I2[lvl-1], res);
}

res.tx /= 2;
res.ty /= 2;

return res;
}

//=====
// motion_translation - do a single for translation only
//=====

#define TRANS_MAT_SIZE 2

fMotion motion_translation (fMonoImg &Ix, fMonoImg &Iy, fMonoImg &It)
{
    double matrix [TRANS_MAT_SIZE*TRANS_MAT_SIZE],
    vect[TRANS_MAT_SIZE], var[TRANS_MAT_SIZE]
    double ix, iy, it;
    fMotion res;
    int x,y,i,j;

    int startX, stopX, startY, stopY;

    // skip the edge of the image
    startX = Ix.GetXmin() + 20;
    startY = Ix.GetYmin() + 20;
    stopX = Ix.GetXmax() - 20;
    stopY = Ix.GetYmax() - 20;

    memset(matrix, 0, sizeof(matrix));
    memset(vect, 0, sizeof(vect));
    memset(var, 0, sizeof(var));

    for (y=startY; y < stopY; y++) {
        for (x=startX; x < stopX; x++) {

```

```

    ix = Ix.Pix(x,y);
    iy = Iy.Pix(x,y);
    it = It.Pix(x,y);

    A [0] = ix;
    A [1] = iy;

    for (i=0; i<TRANS_MAT_SIZE; ++i) {
        vect[i] -= A[i]*it;
        for (j=i; j<TRANS_MAT_SIZE; ++j) {
            matrix[i * TRANS_MAT_SIZE + j] += A[i] * A[j];
        }
    }
}

for (i=1; i<TRANS_MAT_SIZE; i++) {
    for (j=0; j<i; j++) {
        matrix[i * TRANS_MAT_SIZE + j] = matrix[j * TRANS_MAT_SIZE + i];
    }
}

// Solve Equations
solve_2_equations(matrix, vect, var);

res.tx = var[0];
res.ty = var[1];

return res;

/=====
/ motion_rotation - do a single for translation and rotation
/=====

define TRANSROT_MAT_SIZE 3

Motion motion_rotation    (fMonoImg &Ix, fMonoImg &Iy, fMonoImg &It)

```

```

double matrix[TRANSROT_MAT_SIZE*TRANSROT_MAT_SIZE],
vect[TRANSROT_MAT_SIZE], var[TRANSROT_MAT_SIZE]
double ix, iy, it;
double theta;
fMotion res;
int x,y,i,j;

int startX, stopX, startY, stopY;

//skip the edge of the image
startX = Ix.GetXmin() + 20;
startY = Ix.GetYmin() + 20;
stopX = Ix.GetXmax() - 20;
stopY = Ix.GetYmax() - 20;

memset(matrix, 0, sizeof(matrix));
memset(vect, 0, sizeof(vect));
memset(var, 0, sizeof(var));

for (y=startY; y < stopY; y++) {
    for (x=startX; x < stopX; x++) {

        ix = Ix.Pix(x,y);
        iy = Iy.Pix(x,y);
        it = It.Pix(x,y);

        A[0] = x * iy - y * ix;
        A[1] = ix;
        A[2] = iy;

        for (i=0; i<TRANSROT_MAT_SIZE; ++i) {
            vect[i] -= A[i]*it;
            for (j=i; j<TRANSROT_MAT_SIZE; + j) {
                matrix[i * TRANSROT_MAT_SIZE + j] += A[i] * A[j];
            }
        }
    }
}

for (i=1; i<TRANSROT_MAT_SIZE; i++) {
    for (j=0; j<i; j++) {
        matrix[i * TRANSROT_MAT_SIZE + j] = matrix[j * TRANSROT_MAT_SIZE
        + i];
    }
}

II Solve Equations
solve_3_equations(matrix, vect, va

```

Appendix A

```
// Columbia University. Cave lab.
// Generate psf from motion data
//June 2002
```

```
#include "pnm.-env.h"
```

```
#define WIDTH 300
#define HEIGHT 300
```

```
main(int argc, char *argv[])
{
```

```
    fMonoImg res;
    if (argc < 3) {
```

```
        fprintf(stderr,"Usage %s <number of frames> <scale factor> <
        motion_data exit(1);
```

```
    }
```

```
    res = fMonoImg(WIDTH,HEIGHT);
```

```
    res = 0;
```

```
    float tx,ty;
    int x,y;
```

```
    float locx = 0;
    float locy = 0;
```

```
    int n = atoi(argv[1]);
```

```
    float factor = atof(argv[2]);
```

```
    res.Pix(WIDTH/2,HEIGHT/2) ++ ; // first frame
```

```
    for (int i=0; i<n; i++) {
        scanf("%f%f",&tx);
        scanf("%f%f",&ty);
```

```
        tx *= factor;
        ty *= factor;
```

```
        locx += tx;
        locy += ty;
```

```
        printf("%10.6f %10.6f\n",locx,locy);
```

```
        x = (int) (locx + 0.5);
        y = (int) (locy + 0.5);
        res.Pix(x+WIDTH/2,y+HEIGHT/2) ++;
    }

    //normalize

    float max = 1;
    for (int i=0; i<res.GetSize( ); i++) if (res.Pix(i) > max) max = res.Pix(i);
    for (int i=0; i<res.GetSize( ); i++) res.Pix(i) = res.Pix(i) * 255 / max;

    pnm((const bMonoImg) res) .Write("psf.pgm");
```



C:\blur\blur1.m  
June 17, 2002

Page 1  
3:37:35 PM

```
%=====
% The following procedure activates matlab
% Lucy-Richardson deconvolution algorithm used
% for motion deblurring.
% <img.bmp> is the blurred image
% <p.bmp> is the point spread function
% The result is stored in variable 'd'
%
% Columbia university CAVE lab.
% June 2002
%=====
```

```
p=imread('p.bmp','bmp');
p1=double(p);
p1=p1/sum(sum(p1));
```

```
I=imread('img.bmp','bmp');
I1=double(I);
I1=I1/max(max(max(I1)));
I2 = edgetaper(I1,p1);
WT = zeros(size(I2)); WT(5:end-4,5:end-4)= 1;
d=deconvlucy(I2,p1,20,0.1,WT,4);
```

CLAIMS

1. A system for providing a substantially de-blurred image of a scene from a motion blurred image of said scene, comprising:
  - (a) a primary detector for sensing said motion blurred image of said scene at a first predetermined resolution and generating primary image information representing said blurred image;
  - (b) a secondary detector, coupled to said primary detector, for sensing two or more secondary images of said scene at a second predetermined resolution which is less than or equal to said first predetermined resolution, and for generating secondary image information representing said two or more secondary images; and
  - (c) a processor, coupled to said primary detector and receiving said primary image information therefrom, and coupled to said secondary detector and receiving said secondary image information therefrom, adapted to determine motion information from said secondary image information, estimate a point spread function for said motion blurred image from said motion information, and apply said estimated point spread function to said primary image information to thereby generate information representing said substantially de-blurred image.
2. The system of claim 1, further comprising a first camera housing said primary detector, a second camera housing said secondary detector, and a rigid member connecting said first camera to said second camera.
3. The system of claim 1, further comprising a camera housing said primary and said secondary detectors.
4. The system of claim 3, further comprising a beam splitter having one input area and first and second output areas, said beam splitter being optically coupled to said scene at said input area, optically coupled to said primary detector at said first output area, and optically coupled to said secondary detector at said second output area.
5. The system of claim 4, wherein said beam splitter comprises an asymmetric beam splitter adapted to output greater than 50% of an input image through said first output area.
6. The system of claim 5, wherein said beam splitter is adapted to output greater than

80% of an input image through said first output area.

7. The system of claim 4, wherein said beam splitter is adapted to output substantially all visible light corresponding to an input image through said first output area, and at least some non-visible electromagnetic radiation corresponding to an input image through said second output area.

8. The system of claim 3, wherein said primary detector comprises a first portion of a dual-resolution sensor, and said secondary detector comprises a second portion of said dual-resolution sensor.

9. The system of claim 1, wherein said first predetermined resolution is at least twice as great as said second predetermined.

10. A method for providing a substantially de-blurred image of a scene from a motion blurred image of said scene, comprising the steps of:

(a) sensing said motion blurred image of said scene at a first predetermined resolution;

(b) sensing two or more secondary images of said scene at a second predetermined resolution which is less than or equal to said first predetermined resolution while said motion blurred image is sensed;

(c) generating primary image information representing said blurred image and secondary image information representing said two or more secondary images;

(d) determining motion information from said secondary imageS information;

(e) estimating a point spread function for said motion blurred image from said motion information; and

(f) applying said estimated point spread function to said primary image information to thereby generate information representing said substantially de-blurred image.

11. The method of claim 10, wherein said step (b) comprises sensing five or more secondary images of said scene at said second predetermined resolution, and said step (c) comprises generating primary image information representing said blurred image and secondary image information representing said five or more secondary images.

12. The method of claim 10, wherein step (d) comprises the step of determining global motion information from said secondary image information.
13. The method of claim 12, wherein step (e) comprises the step of estimating a continuous point spread function for said motion blurred image from said global motion information.
14. The method of claim 13, wherein step (f) comprises the step of deconvolving said primary image information with estimated point spread function to thereby generate information representing said substantially de-blurred image.
15. A method for providing a substantially de-blurred image of a scene from primary image information representing a motion blurred image of said scene captured at a first predetermined resolution, and secondary image information representing two or more secondary images of said scene captured at a second predetermined resolution which is less than or equal to said first predetermined resolution, comprising the steps of:
- (a) determining motion information from said secondary image information;
  - (b) estimating a point spread function for said motion blurred image from said motion information; and
  - (c) applying said estimated point spread function to said primary image information to thereby generate information representing said substantially de-blurred image.
16. The method of claim 16, wherein step (a) comprises the step of determining global motion information from said secondary image information.
17. The method of claim 17, wherein step (b) comprises the step of estimating a continuous point spread function for said motion blurred image from said global motion information.
18. The method of claim 18, wherein step (c) comprises the step of deconvolving said primary image information with estimated point spread function to thereby generate information representing said substantially de-blurred image.
19. A method for estimating a point spread function for an image which was blurred due to motion of an image sensor which captured said image, comprising the steps of:

- (a) receiving discrete motion information indicative of said motion of said image sensor;
- (b) determining a continuous two-dimensional path corresponding to said discrete motion points by interpolating missing data;
- (c) partitioning said determined continuous two-dimensional path into a plurality of frames; and
- (d) constructing equal-area regions within each of said partitioned frames to determine a mean energy value for each of said frames.

20. The method of claim 19, wherein step (b) comprises the step of determining said continuous two-dimensional path corresponding to said discrete motion points by Spline interpolation.

21. The method of claim 19, wherein step (c) comprises the step of partitioning said determined continuous two-dimensional path into said plurality of frames by Voronoi Tessellation.

22. The method of claim 19, further comprising the step of normalizing said determined mean energy values.

23. A method for interactively determining a de-blurred image corresponding to a motion blurred image of a scene, comprising the steps of:

- (a) selecting one or more features within said motion blurred image;
- (b) forming one or more de-blurred image feature models, each of which corresponds to one of said one or more selected features;
- (c) estimating a point spread function using said selected one or more features and said corresponding one or more de-blurred image feature models; and
- (d) determining said de-blurred image using said estimated point spread function and said motion blurred image.

24. The method of claim 23, wherein at least one of said one of said one or more selected features comprises a point, and said step (b) comprises the step of using a point tool to form at least one of said one or more de-blurred image feature models.

25. The method of claim 23, wherein at least one of said one of said one or more selected features comprises a line, and said step (b) comprises the step of using a line tool to form at least one of said one or more de-blurred image feature models.
26. The method of claim 23, wherein at least one of said one of said one or more selected features comprises a ramp, and said step (b) comprises the step of using a ramp tool to form at least one of said one or more de-blurred image feature models.
27. The method of claim 23, wherein at least one of said one of said one or more selected features comprises a corner, and said step (b) comprises the step of using a corner tool to form at least one of said one or more de-blurred image feature models.
28. A method for measuring a point spread function for an image of a scene which was blurred due to motion of an image sensor which captured said image, comprising the steps of:
- (a) aiming a stabilized light beam at said scene to thereby form a light point in said scene;
  - (b) capturing a blurred image of said scene, including a blurred image of said light point, using a non-stabilized image sensor; and
  - (c) determining said point spread function from said blurred image of said light point.
29. The method of claim 28, wherein said stabilized light beam originates from a laser attached to said non-stabilized image sensor.
30. The method of claim 28, wherein said stabilized light beam originates from a laser positioned independently from said non-stabilized image sensor.
31. The method of claim 28, further comprising the step of generating a de-blurred image corresponding to said blurred image of said scene.

1/12

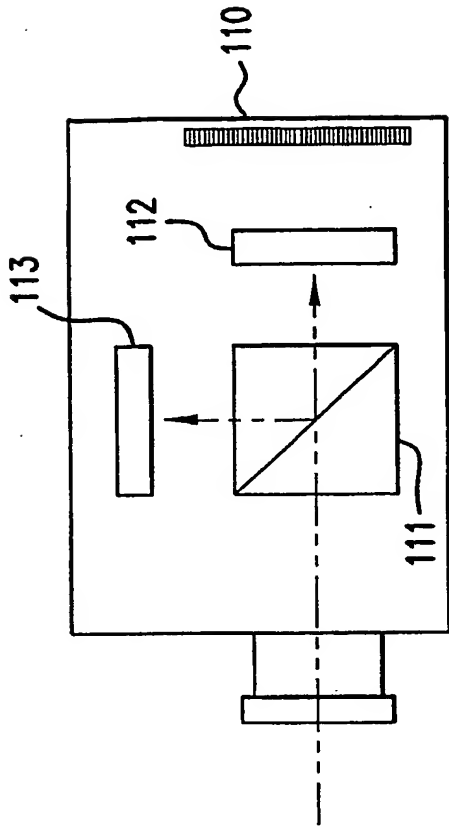


FIG. 1B

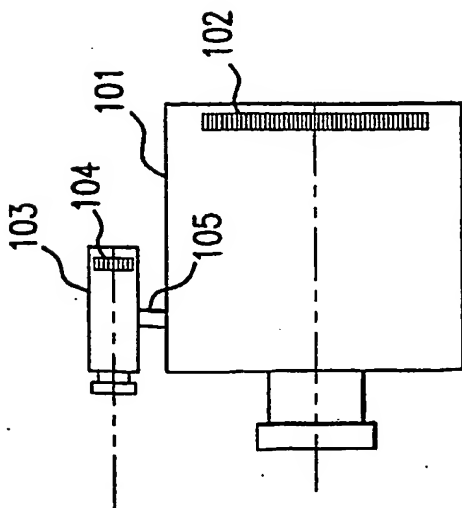


FIG. 1A

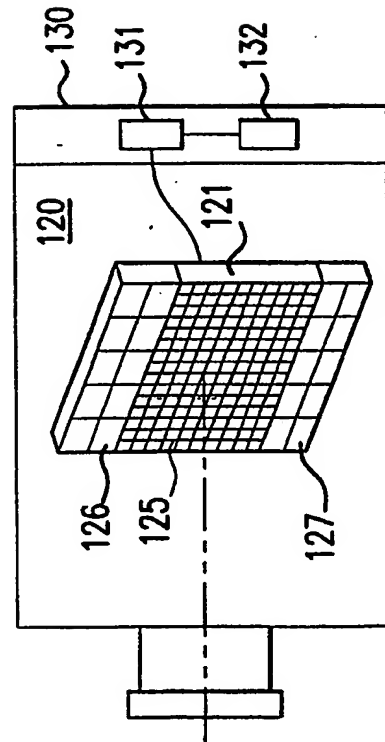


FIG. 1C

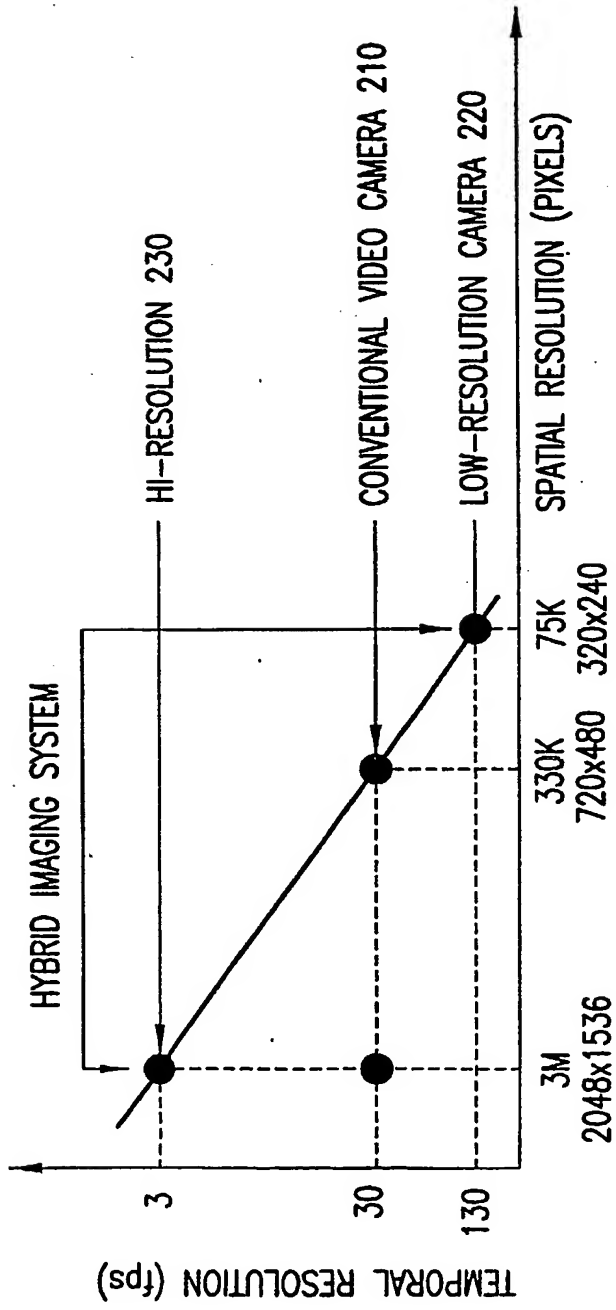


FIG.2



3/12

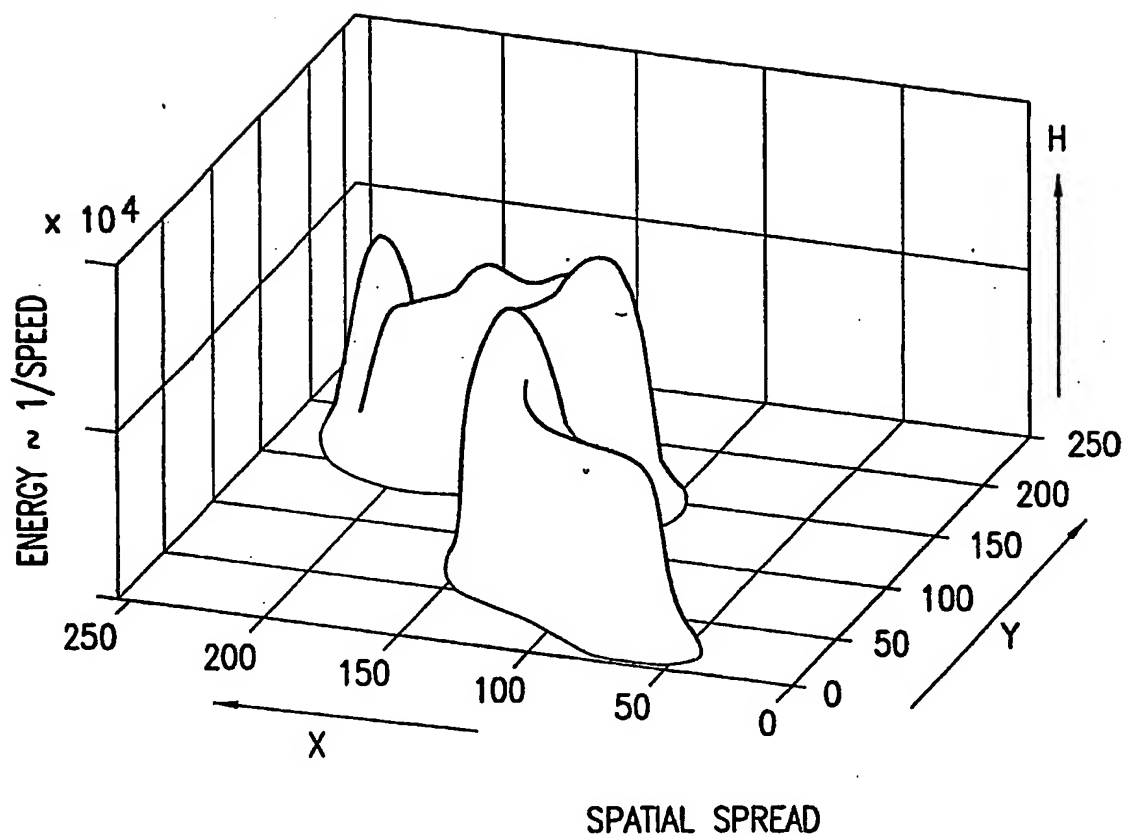


FIG.3

4/12

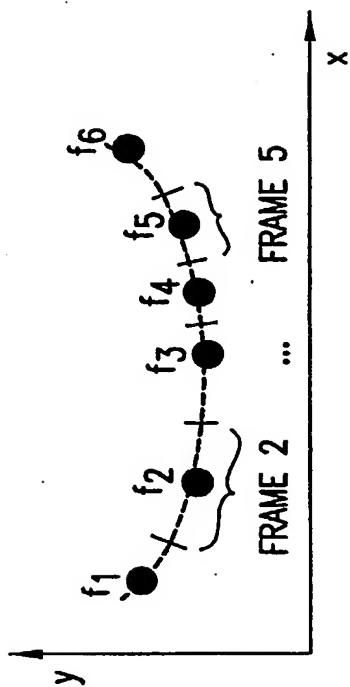


FIG. 4B

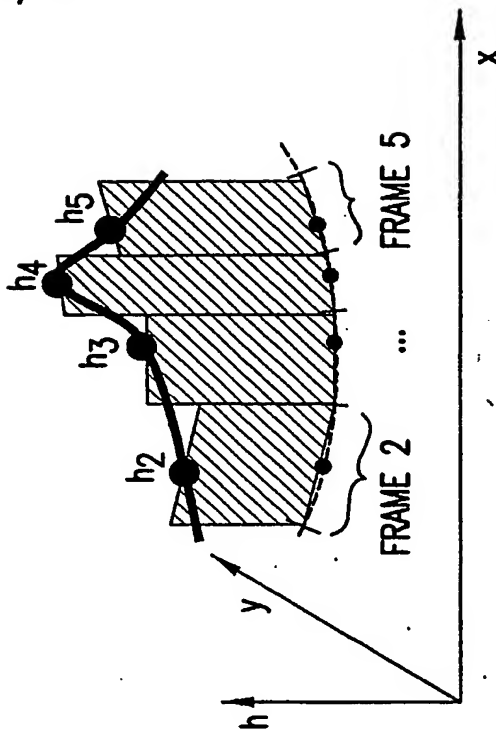


FIG. 4D

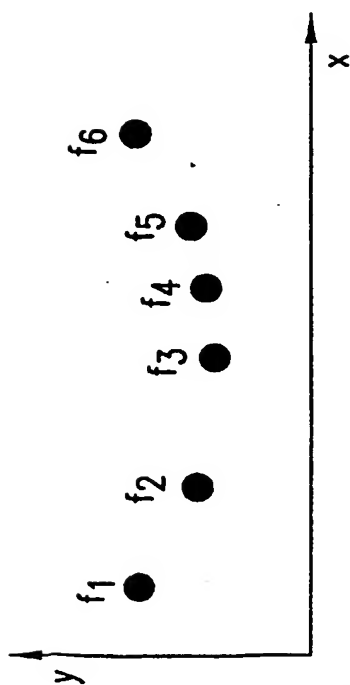


FIG. 4A

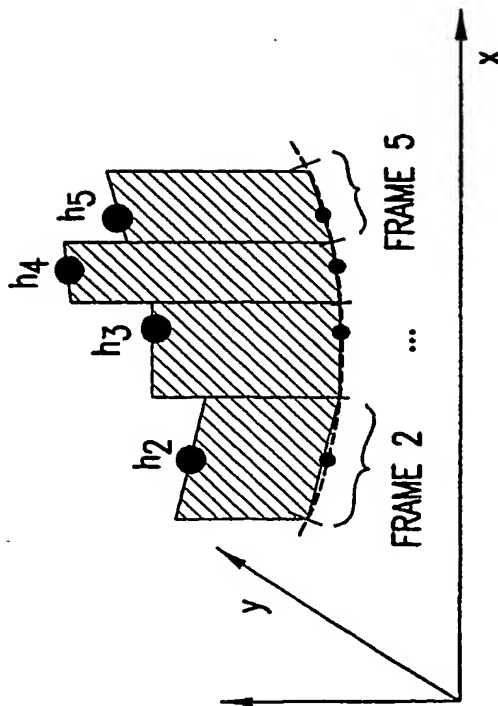


FIG. 4C

5/12

## FLOW DIAGRAM HYBRID IMAGING

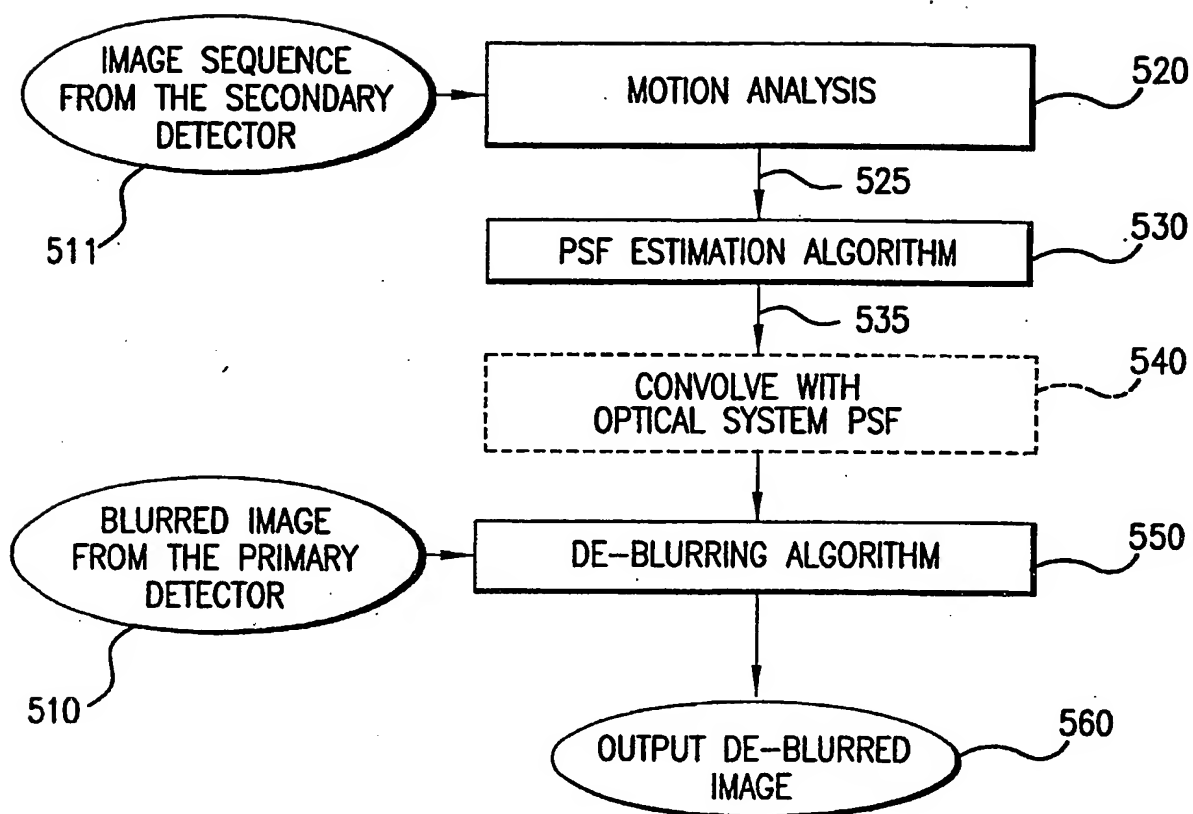


FIG.5A

6/12

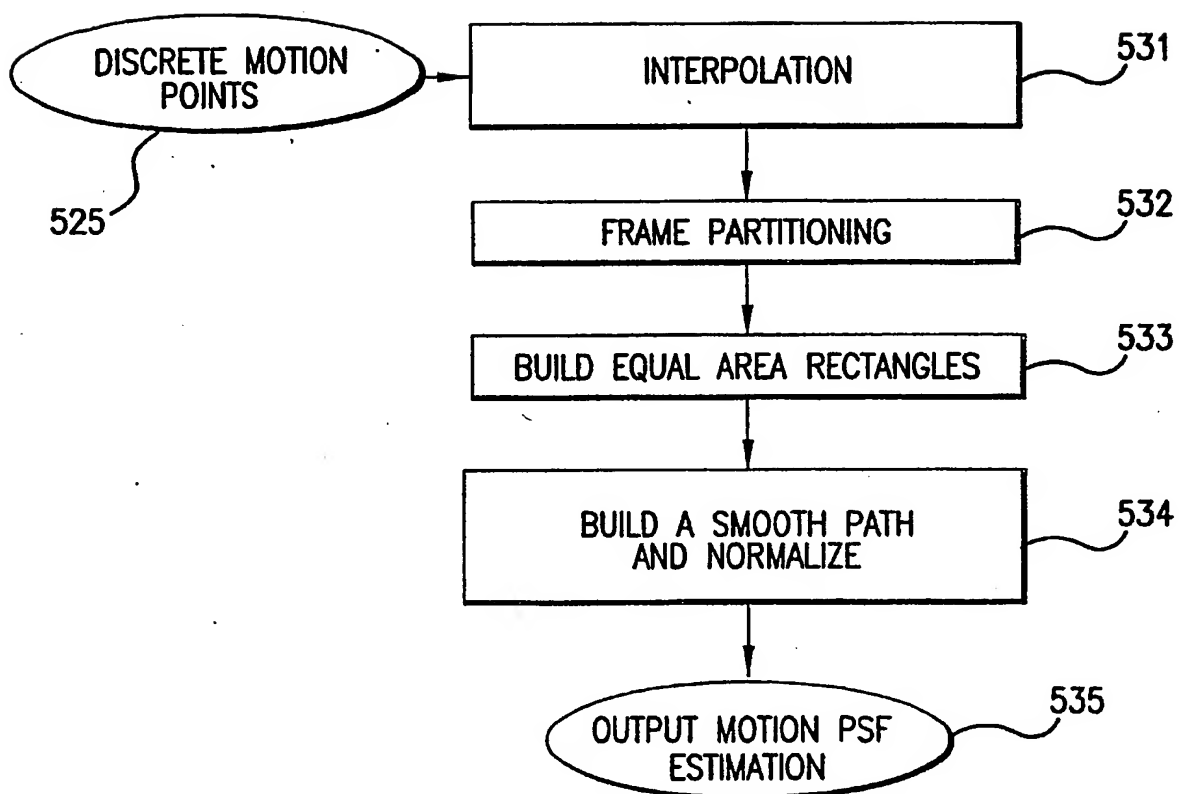


FIG.5B

7/12

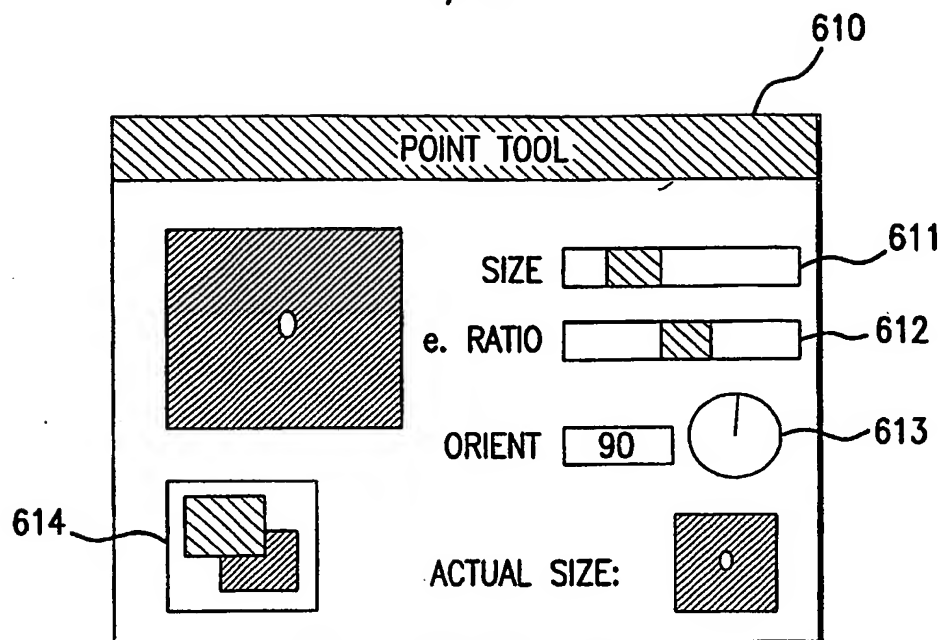


FIG. 6A

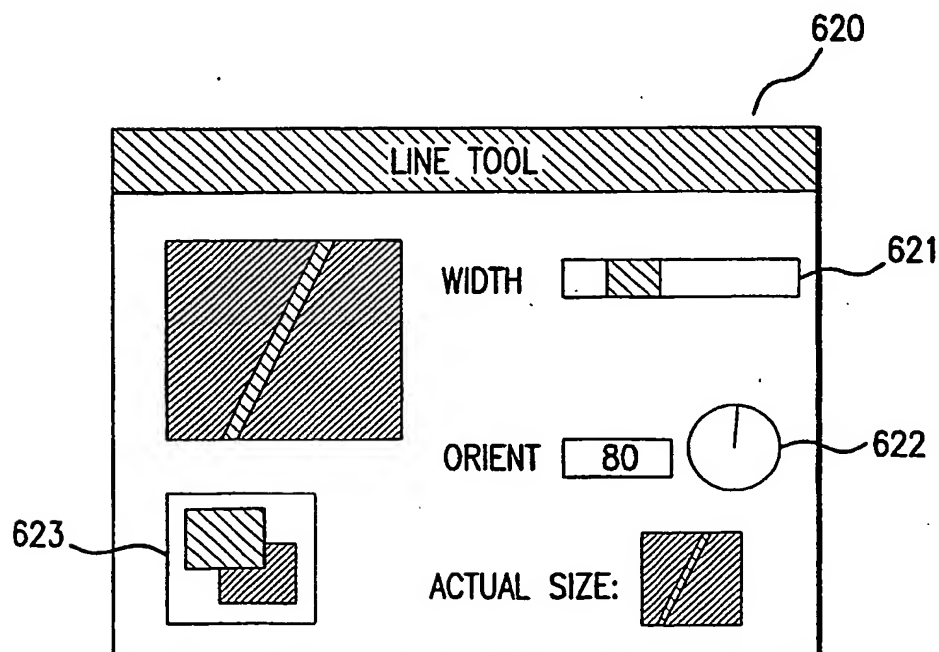


FIG. 6B

8/12

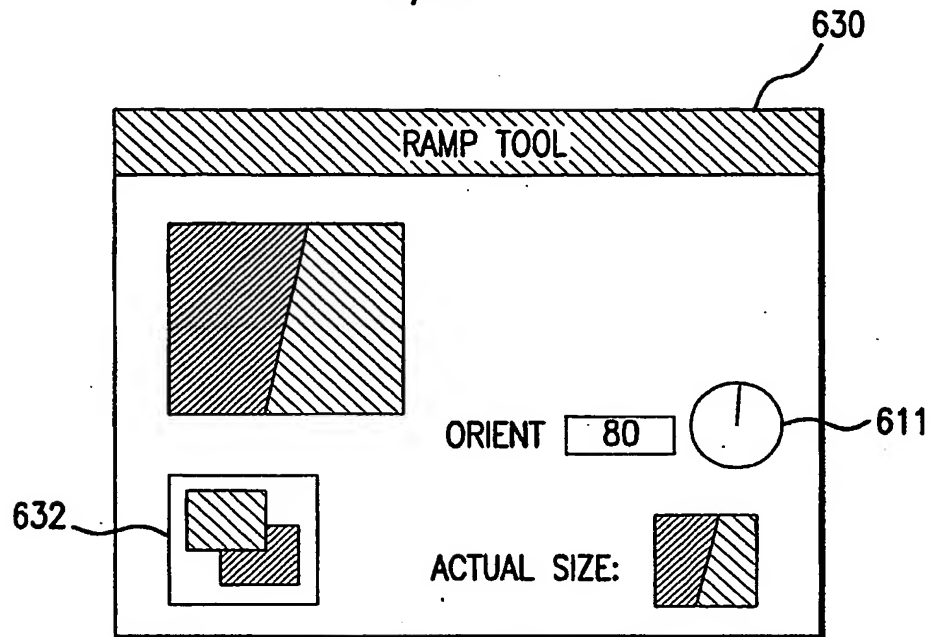


FIG. 6C

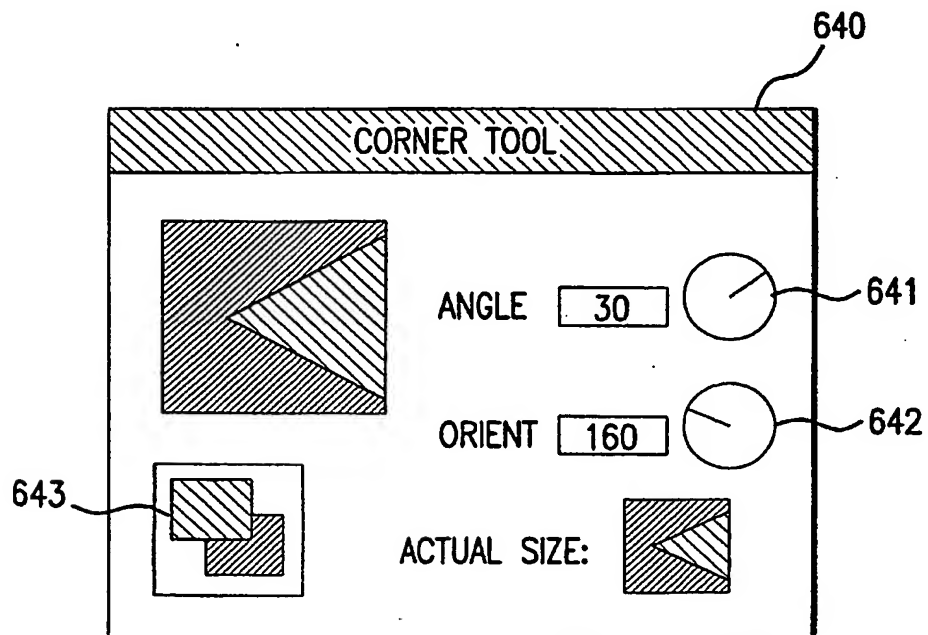


FIG. 6D

9/12

## INTERACTIVE PSF ESTIMATION

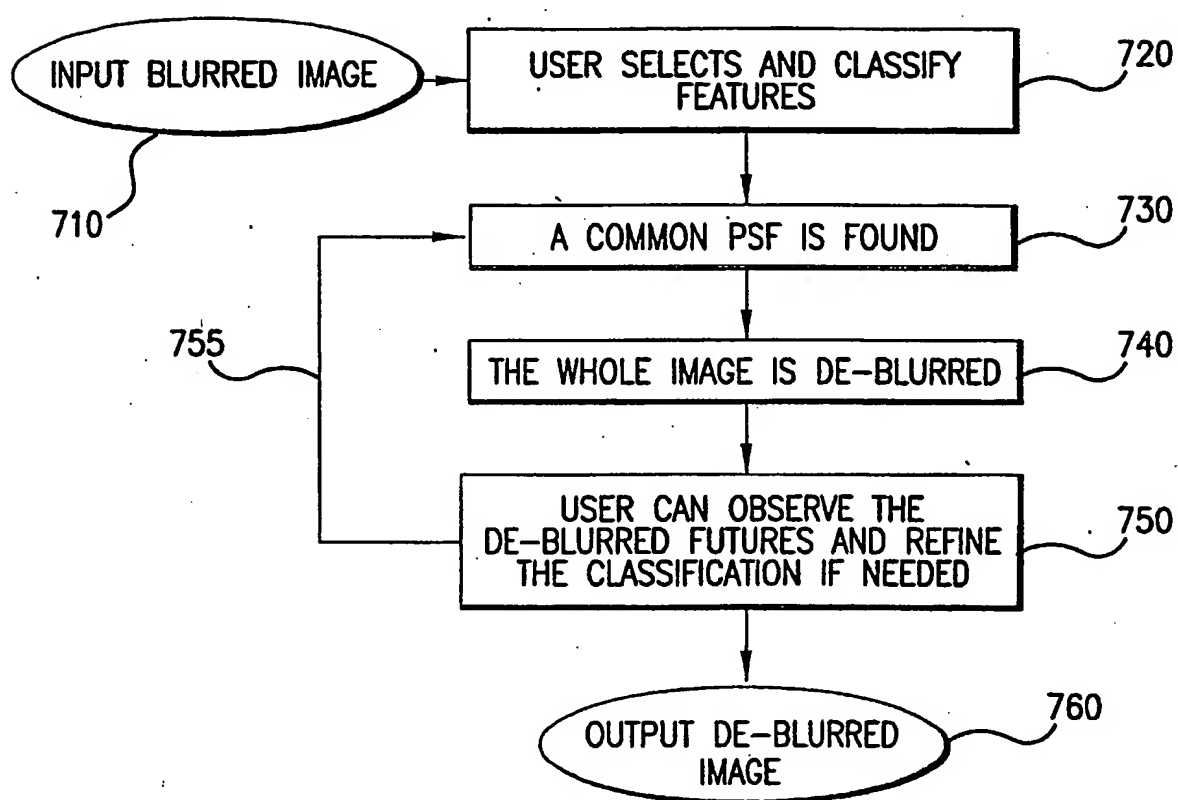


FIG.7

10/12

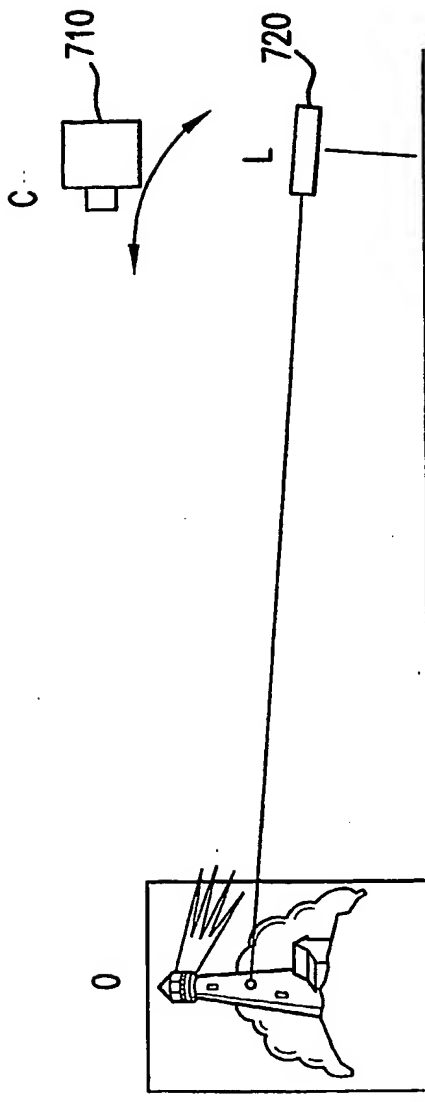


FIG. 8A

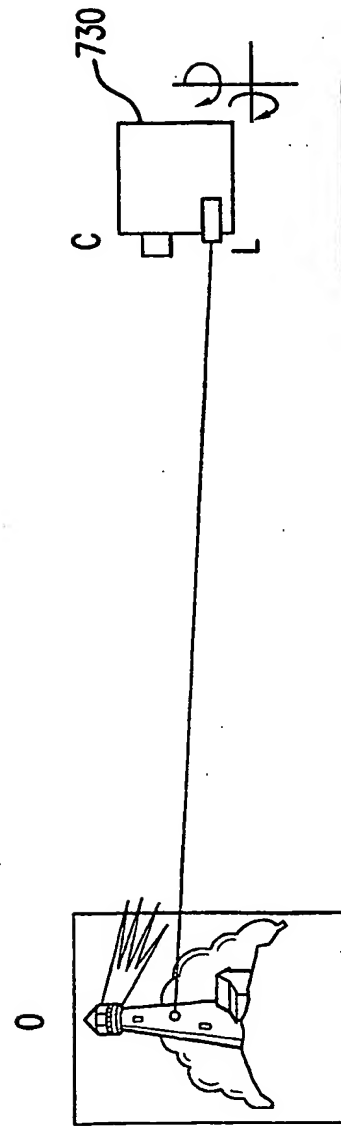
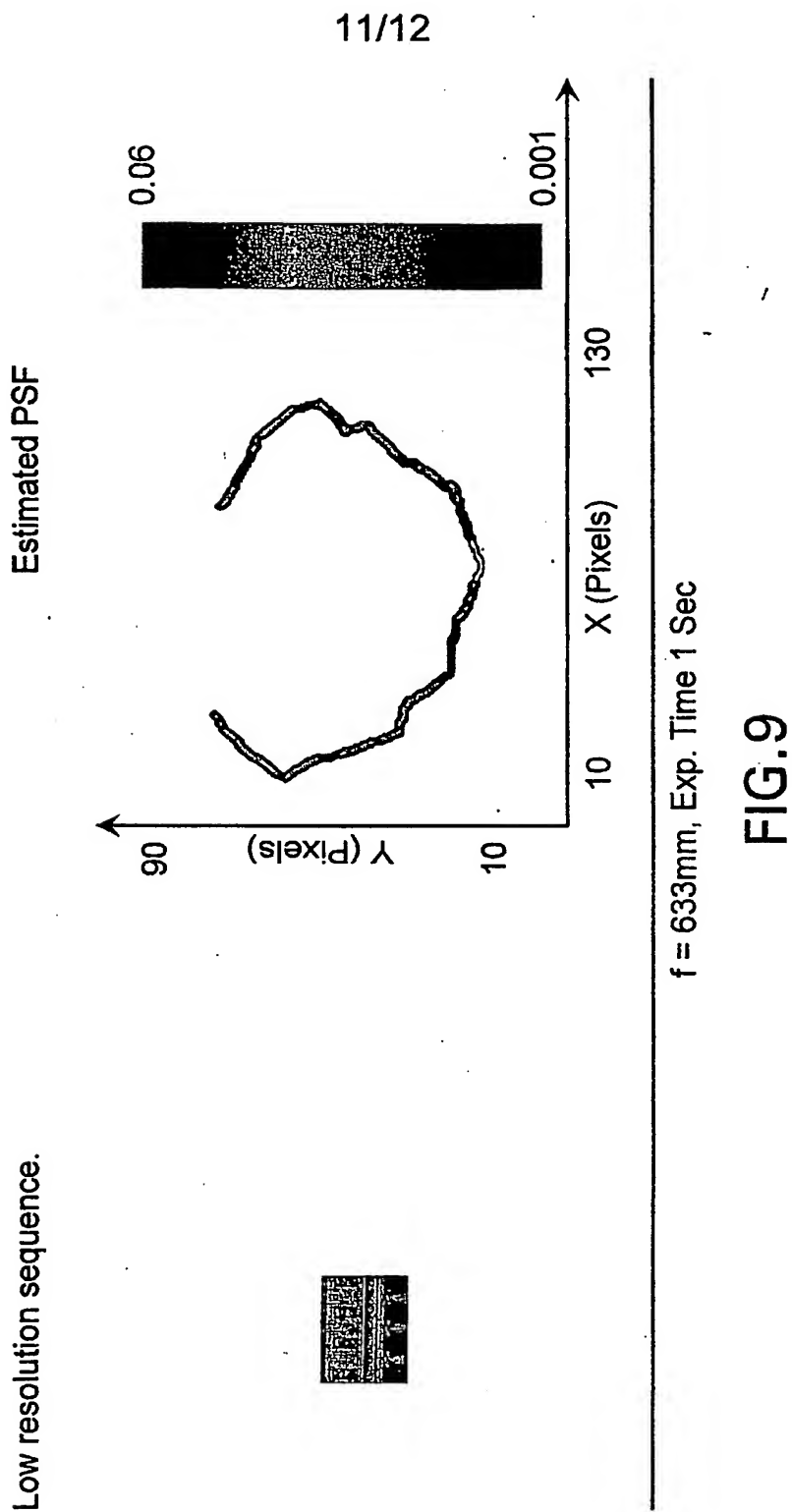
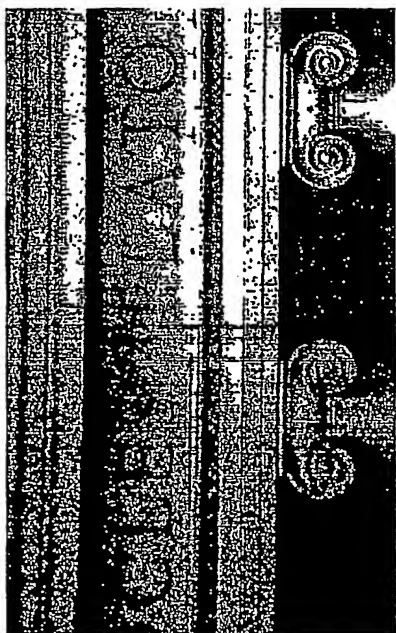


FIG. 8B



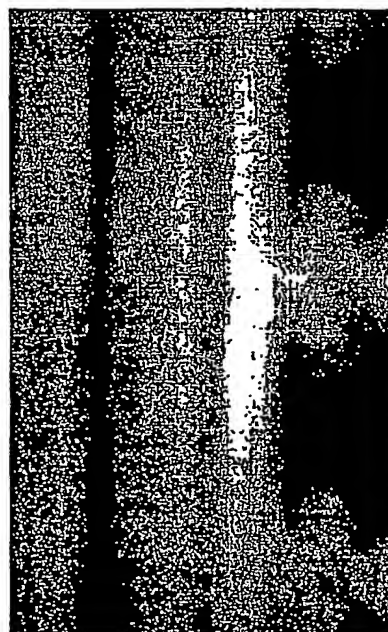


12/12



Tripod image

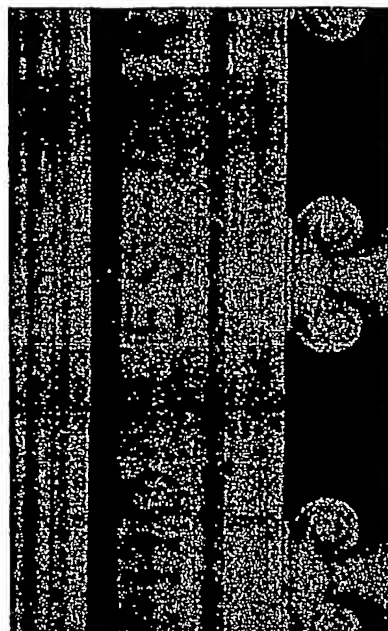
FIG. 10A



Blurred image

f = 633mm, Exp. Time 1 Sec

FIG. 10B



Deblurred image

f = 633mm, Exp. Time 1 Sec

FIG. 10C